REMARKS

Claims 1-5, 8-13, 16, 18, 19, 22, 24, 25, 27-30 are pending. Claims 6, 7, 14, 15, 17, 20, 21, 23, 26, and 31 have been cancelled. Claims 1, 8, 16, 22, and 28 have been amended. No new matter has been added. Reexamination and reconsideration of the present application are respectfully requested.

In the July 12, 2005 Office Action, the Examiner rejected claims 1-5, 7-13, 15-25, 27-30 under 35 U.S.C. § 103(a) as being unpatentable over Krishna (WO 01/05086) in view of Hausman et al. (US Patent No. 6,112,252). The Examiner rejected claims 6, 14, 26 and 31 under 35 U.S.C. 103(a) as being unpatentable over Krishna, Hausman, and Kagan et al. (US Patent No. 6,243,787).

Claim 1, as amended, recites:

> An encrypted network system, comprising:
> a network to transmit an encrypted packet; and
> a computer to receive said encrypted packet from said network, and to perform a decryption operation thereupon to convert said encrypted packet to a decrypted packet, said computer including:
> a network interface to provide electronic communication between said computer and said network,
> a network driver to regulate said decryption operation;
> a controller to perform said decryption operation;
> a host memory to store data that is used or generated by said decryption operation, and
> a bus providing electronic communication among said network interface, said network driver, said host memory and said controller, said controller asserting an interrupt prior to a complete transfer of said decrypted packet from said controller to said host memory,
> wherein said controller asserts an additional interrupt after completion of said decryption operation, and said network driver specifies an average latency value to said controller for use in said decryption operation.

The Examiner stated that Krishna and Hausman don't clearly teach that a controller asserts an additional interrupt after completion of said decryption operation.

However, the Examiner stated that Kagan teaches "asserting an interrupt after completion of operation [**Col. 2 lines 54-55 'after sending the data, the peripheral device assert an interrupt' Col. 2 lines 65-67 'the host interface will receive the interrupt packet only after it has received all of the preceding data packets.']."** (emphasis in original). The Examiner further stated that it would be obvious to one of ordinary skill in the art at the time the invention was made to assert an interrupt after sending the data. The Examiner stated that such modification would be obvious because one of ordinary skill in the art would be motivated to use packet switching fabrics to connect a computer host to peripheral devices to reduce latency and processing time required for servicing of interrupts by the CPU. In support, the Examiner cited Kagan, Col. 2 lines 45-48, which states "... a method and system for communicating between a CPU and peripheral devices via a switching fabric that reduces latency and processing time required for servicing of interrupts by the CPU."

Kagan teaches that a "peripheral device" asserts an interrupt after the "sending of data." Kagan does not teach a *controller* asserting an additional interrupt after completion of a *decryption operation*, as recited by amended claim 1. Specifically, Kagan teaches "[a]fter sending the data, the peripheral device asserts an interrupt," Col. 2, line 54, and "the host interface will receive the interrupt packet only after it has received all of the preceding data packets," Col. 2, line 66. Importantly, the interrupt in Kagan is delivered "via a switching fabric." Col. 2, line 42.

Generally, encrypted data and benign interrupts may be sent across network fabrics. However, in high-security decryption operations such as the present invention, it is important to insulate sensitive decrypted data and related interrupts from volatile networks. Thus, sending an interrupt across the network, as taught by Kagan, is wholly unworkable. The present invention offers a different approach specifically tailored to the demands of high-security systems.

A peripheral device, by definition, is external to the computer. It is connected to the computer across a switching fabric. Kagan, Col. 2, Lines 45-48. Thus, an interrupt sent from a peripheral device, as taught by Kagan, must travel across the network to reach the target computer. In contrast, amended claim 1 teaches that an interrupt is asserted by a controller that is included in the computer. ("said computer including: ... a controller to perform said decryption operation...<u>wherein said controller asserts an additional interrupt after completion of said decryption operation</u>"). In the present invention, the encryption operations are included in the computer according to the following structure and sequence: (a) a network to transmit the encrypted data packet, (b) a computer to receive the encrypted data packet, (c) a controller to perform the decryption operation, and (d) the controller asserting an interrupt after completion of decryption operation (amended claim 1). Thus, the interrupt is sent by the controller included within the computer, and not across the network fabric.

This results in greater security because an interrupt sent by a controller *internal* to the computer need not travel across the network, where security may be readily compromised through misdirected packets or malicious security breaches. In the present invention, decrypted data and related interrupts must be insulted from the network.

Kagan's approach of asserting interrupts across volatile networks may be suitable when connecting printers or other peripherals to target computers. But it is wholly unworkable in security decryption applications. Sending decryption interrupts across a network fabric would compromise security. The present invention improves upon Kagan and the other cited art by keeping interrupts *internal* to the computer, and only asserting interrupts between the internal components across a secure bus. The decryption operation is thereby insulated from the network at large, resulting in substantially improved security compared to the system disclosed by Kagan.

It would not have been obvious to one skilled in the art at the time the invention was made to "modify Kagan into the teaching of Krishna and Hausman to assert an interrupt after sending the data." First, as noted above, Kagan's approach of sending an interrupt across a volatile network is wholly unworkable in the context of high-security applications. In Kagan, the purpose of sending the interrupt packet is to "synchronize data and interrupt handling" and to "make sure that the data have been completely written to the memory before the CPU attempts to read it." Col. 1, line 50. Ensuring security was not among the intended purposes of Kagan. Thus, it would not have been obvious to modify Kagan's unworkable low-security synchronization approach for the present invention's high-security context because Kagan teaches sending interrupts from a peripheral device across the switching fabric, not between internal computer components across a high-security system bus.

Second, amended claim 1 recites a controller "asserting an interrupt prior to a complete transfer of said decrypted packet" and again "after completion of said decryption operation" (emphasis added). It does not, however, recite assertion of an interrupt after *sending of data*, as taught by Kagan. It would not have been obvious to modify Kagan to send an interrupt prior to complete transfer or after completing the decryption operation, as recited by amended claim 1, because no decryption operations take place on a peripheral device. Even if decryption operations did take place on peripheral devices, security demands that decryption interrupts not be sent over a volatile switching fabric.

Claim 1 also recites "wherein said network driver specifies an *average latency value* to said controller for use in said decryption operation." The Examiner stated that Krishna discloses that "the classification engine provides support for general IPSec policy rule sets, including wild cards, overlapping rules, conflicting rules and conducts deterministic searches in a fixed number of clock cycles" (page 12, lines 31-33). The classification engine disclosed in Krishna determines security association information required for processing packets. Page 11, line 20. It performs lookups from databases stored in associated memory, and employs IPSec (see RFC2401) to provide security at the IP layer. Page 12, line 31.

IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec supports two encryption modes: transport and tunnel. Transport mode encrypts only the data portion (payload) of each packet, but leaves the header untouched. The more secure Tunnel mode encrypts both the header and the payload. On the receiving side, an IPSec-compliant device decrypts each packet. IPSec policy rule sets are used to tell the operating system kernel which individual packets require security. Rule sets may use wild cards to specify any parameter not explicitly stated. Overlapping or conflicting rules are a result of poor security management, and can cause a firewall to mistakenly drop traffic that should be allowed. The problem of overlapping or conflicting rules can be remedied by a consistent site-wide rule policy. Finally, deterministic searches can improve response time where extensive rule sets are employed.

The IPSec and related policy rule sets do not disclose average latency values. Average latency values specify the latencies of various operations such as receiving a packet, transferring to host, handling interrupts, and parsing packets. Specification, p. 7. IPSec, on the other hand, is directed to providing an open data confidentiality, data integrity, and data authentication framework. Wildcards and deterministic searches are tools used by IPSec to facilitate the processing of packets at the IP layer, and overlapping or conflicting rules present a problem that can generally be remedied through consistent security management. Thus, the classification engine and related IPSec policy rule sets disclosed in Krishna do not disclose a network driver specifies an average latency value to said controller for use in said decryption operation, as recited by amended claim 1.

For the foregoing reasons, Applicant respectfully submits that amended claim 1 is allowable over the cited art. Applicant has amended independent claims 8, 16, 22, and 28 to recite the additional interrupt and average latency values substantially as recited in claim 1. Thus, Applicant respectfully submits that amended claims 8, 16, 22 and 28 are allowable over the cited art for the same reasons discussed above with regard to amended claim 1.

Claims 2-5 depend, directly or indirectly, upon independent claim 1. Thus, Applicant respectfully submits that dependent claims 2-5 are allowable over the cited art for the same reasons discussed above with regard to amended claim 1. Claims 9-13 depend, directly or indirectly, upon independent claim 8. Thus, Applicant respectfully submits that dependent claims 9-13 are allowable over the cited art for the same reasons discussed above with regard to amended claim 8. Claims 18, 19 and 21 depend directly upon independent claim 16. Thus, Applicant respectfully submits that dependent claims 18, 19 and 21 are allowable over the cited art for the same reasons discussed above with regard to amended claim 16. Claims 24, 25 and 27 depend, directly or indirectly, upon independent claim 22. Thus, Applicant respectfully submits that dependent claims 24, 25 and 27 are allowable over the cited art for the same reasons discussed above with regard to amended claim 22. Claims 29 and 30 depend, directly or indirectly, upon independent claim 28. Thus, Applicant respectfully submits that dependent claims 29 and 30 are allowable over the cited art for the same reasons discussed above with regard to amended claim 28.

Applicant believes that the foregoing remarks place the application in condition for allowance, and a favorable action is respectfully requested. If for any reason the Examiner finds the application other than in condition for allowance, the Examiner is requested to call either of the undersigned attorneys at the Los Angeles telephone number (213) 488-7100 to discuss the steps necessary for placing the application in condition for allowance should the Examiner believe that such a telephone conference would advance prosecution of the application.

Respectfully submitted,

PILLSBURY WINTHROP LLP

Date: November 14, 2005

By: Roger R. Wise
Roger Wise
Registration No. 31,204
Attorney for Applicant(s)

By:
Ryan E. Hatch
Registration No. 55,252
Attorney for Applicant(s)

725 South Figueroa Street, Suite 2800
Los Angeles, CA 90017-5406
Telephone: (213) 488-7100
Facsimile: (213) 629-1033